Access List Commands

This module describes the Cisco IOS XR software commands used to configure IP Version 4 (IPv4) and IP Version 6 (IPv6) access lists.

For detailed information about ACL concepts, configuration tasks, and examples, refer to the *IP Addresses and Services Configuration Guide for Cisco 8000 Series Routers*.

clear access-list ipv4

To clear IPv4 access list counters, use the **clear access-list ipv4** command in XR EXEC mode.

clear access-list ipv4 *access-list-name* **hardware** {**clear access-list ipv4** *access-list-name* **hardware** {**ingress** | **egress** } [**interface** *interface-path-id*] [**sequence** *sequence-number*] [**location** *node-id*] }

Syntax Description

access-list-name	Name of a particular IPv4 access list. The name cannot contain a spaces or quotation marks, but can include numbers.
sequence-number	(Optional) Specific sequence number with which counters are cleared for an access list. Range is 1 to 2147483644.
ingress	Specifies an inbound direction.
egress	Specifies an outbound direction.
interface-path-id	Physical interface or virtual interface. **Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	(Optional) Clears hardware resource counters from the designated node. The *node-id* argument is entered in the *rack/slot/module* notation.

Command Default

The default clears the specified IPv4 access list.

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

Use the **clear access-list ipv4** command to clear counters for a specified configured access list. Use a sequence number to clear counters for an access list with a specific sequence number.

Use an asterisk (*****) in place of the *access-list-name* argument to clear all access lists.

Task ID

Task ID	Operations
basic-services	read, write
acl	read, write
bgp	read, write, execute

Examples

In the following example, counters for an access list named *marketing* are cleared:

```
Router# show access-lists ipv4 marketing hardware ingress location 0/RP0/CPU0
ipv4 access-list marketing
10 permit ipv4 192.168.34.0 0.0.0.255 any
20 permit ipv4 172.16.0.0 0.0.255.255 any
30 deny tcp host 172.16.0.0 eq 2330 host 192.168.202.203 (23345 matches)

Router# clear access-list ipv4 marketing hardware ingress location 0/RP0/CPU0
```

clear access-list ipv6

To clear IPv6 access list counters, use the **clear access-list ipv6** command in .

clear access-list ipv4 *access-list-name* **hardware** {**ingress** | **egress** } [**interface** *interface-path-id*] [**sequence** *sequence-number*] [**location** *node-id*]

Syntax Description

access-list-name	Name of a particular IPv6 access list. The name cannot contain a spaces or quotation marks, but can include numbers.
sequence-number	(Optional) Specific sequence number for a particular access control entry (ACE) with which counters are cleared for an access list. Range is 1 to 2147483644.
ingress	(Optional) Specifies an inbound direction.
egress	(Optional) Specifies an outbound direction.
interface-path-id	Physical interface or virtual interface. **Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (?) online help function.
location *node-id*	(Optional) Clears counters for an access list enabled on a card interface. The *node-id* argument is entered in the rack/slot/module notation.

Command Default

The default clears the specified IPv6 access list.

Command Modes

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

The **clear access-list ipv6** command is similar to the **clear access-list ipv4** command, except that it is IPv6-specific.

Use the **clear access-list ipv6** command to clear counters for a specified configured access list. Use a sequence number to clear counters for an access list with a specific sequence number

Use an asterisk (*****) in place of the *access-list-name* argument to clear all access lists.

Task ID

Task ID	Operations
basic-services	read, write
acl	read, write

Task ID	Operations
network	read, write

Examples

In the following example, counters for an access list named *marketing* are cleared:

```
Router# show access-lists ipv6 marketing hardware ingress location 0/RP0/CPU0
ipv6 access-list marketing
  10 permit ipv6 3333:1:2:3::/64 any
  20 permit ipv6 4444:1:2:3::/64 any
  30 permit ipv6 5555:1:2:3::/64 any
Router# clear access-list ipv6 marketing hardware ingress location 0/RP0/CPU0
```

copy access-list ipv4

To create a copy of an existing IPv4 access list, use the **copy access-list ipv4** command in XR EXEC mode.

copy access-list ipv4 *source-acl destination-acl*

Syntax Description

source-acl	Name of the access list to be copied.
destination-acl	Name of the destination access list where the contents of the *source-acl* argument is copied.

Command Default

None

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

Use the **copy access-list ipv4** command to copy a configured access list. Use the *source-acl* argument to specify the access list to be copied and the *destination-acl* argument to specify where to copy the contents of the source access list. The *destination-acl* argument must be a unique name; if the *destination-acl* argument name exists for an access list or prefix list, the access list is not copied. The **copy access-list ipv4** command checks that the source access list exists then checks the existing list names to prevent overwriting existing access lists or prefix lists.

Task ID

Task ID	Operations
acl	read, write
filesystem	execute

Examples

In the following example, a copy of access list list-1 is created:

```
Router# show access-lists ipv4 list-1

ipv4 access-list list-1
  10 permit tcp any any log
  20 permit ip any any
Router# copy access-list ipv4 list-1 list-2
Router# show access-lists ipv4 list-2
ipv4 access-list list-2
  10 permit tcp any any log
  20 permit ip any any
```

In the following example, copying the access list list-1 to list-3 is denied because a list-3 access list already exists:

```
Router# copy access-list ipv4 list-1 list-3

list-3 exists in access-list

Router# show access-lists ipv4 list-3

ipv4 access-list list-3
  10 permit ip any any
  20 deny tcp any any log
```

copy access-list ipv6

To create a copy of an existing IPv6 access list, use the **copy access-list ipv6** command in .

copy access-list ipv6 *source-acl destination-acl*

Syntax Description

source-acl	Name of the access list to be copied.
destination-acl	Destination access list where the contents of the *source-acl* argument is copied.

Command Default

No default behavior or value

Command Modes

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

Use the **copy access-list ipv6** command to copy a configured access list. Use the *source-acl* argument to specify the access list to be copied and the *destination-acl* argument to specify where to copy the contents of the source access list. The *destination-acl* argument must be a unique name; if the *destination-acl* argument name exists for an access list or prefix list, the access list is not copied. The **copy access-list ipv6** command checks that the source access list exists then checks the existing list names to prevent overwriting existing access lists or prefix lists.

Task ID

Task ID	Operations
acl	read, write
filesystem	execute

Examples

In this example, a copy of access list list-1 is created:

```
Router# show access-lists ipv6 list-1

ipv6 access-list list-1
  10 permit tcp any any log
  20 permit ipv6 any any

Router# copy access-list ipv6 list-1 list-2

Router# show access-lists ipv6 list-2

ipv6 access-list list-2
  10 permit tcp any any log
  20 permit ipv6 any any
```

In this example, copying access list list-1 to list-3 is denied because a list-3 access list already exists:

```
Router# copy access-list ipv6 list-1 list-3

list-3 exists in access-list

Router# show access-lists ipv6 list-3
ipv6 access-list list-3
  10 permit ipv6 any any
  20 deny tcp any any log
```

deny (IPv4)

To set conditions for an IPv4 access list, use the **deny** command in access list configuration mode. There are two versions of the **deny** command: **deny** (source), **deny** (destination), and **deny** (protocol). To remove a condition from an access list, use the **no** form of this command.

[*sequence-number*] **deny** *source* [*source-wildcard*] **[log** | | **log-input]**
[*sequence-number*] **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp* [**bitmask** *value*]] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
no *sequence-number*

Internet Control Message Protocol (ICMP)
[*sequence-number*] **deny icmp** *source source-wildcard destination destination-wildcard* [*icmp-type*] [*icmp-code*] [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [**log**][**icmp-off**]

Transmission Control Protocol (TCP)
[sequence-number] **permit tcp** { *source-ipv4-prefix/ prefix-length* | *any* | *host source-ipv4-address ipv4-wildcard-mask/prefix-length* } [*operator* { *port* | *protocol-port* }] { *destination-ipv4-prefix/ prefix-length* | *any* | *host destination-ipv4-address ipv4-wildcard-mask/prefix-length* } [*operator* { *port* | *protocol* | *port* }] [**dscp** *value*] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**established**] { **match-any | match-all | + | -** } [*flag-name*] [**log**]

Internet Group Management Protocol (IGMP)
[*sequence-number*] **deny igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**dscp** *value*] [**fragments**] [**log**]

User Datagram Protocol (UDP)
[*sequence-number*] **deny udp** *source source-wildcard* [*operator* {*portprotocol-port*}] *destination destination-wildcard* [*operator* {*portprotocol-port*}] [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [**log**]

Syntax Description

sequence-number	(Optional) Number of the **deny** statement in the access list. This number determines the order of the statements in the access list. The number can be from 1 to 2147483644. (By default, the first statement is number 10, and the subsequent statements are incremented by 10.)
source	Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *source* combination as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.

source-wildcard	Wildcard bits to be applied to the source. There are three alternative ways to specify the source wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *source* combination as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
protocol	Name or number of an IP protocol. It can be one of the keywords **ahp** , **esp** , **gre** , **icmp** , **igmp** , **igrp** , **ip** , **ipinip** , **nos** , **ospf** , **pim** , **pcp** , **tcp** , or **udp** , or an integer from 0 to 255 representing an IP protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the **ip** keyword. ICMP, and TCP allow further qualifiers, which are described later in this table. **Note** Filtering on AHP protocol is not supported.
destination	Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for the *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *destination* combination as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
destination-wildcard	Wildcard bits to be applied to the destination. There are three alternative ways to specify the destination wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the **any** keyword as an abbreviation for a *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *destination* combination as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
precedence *precedence*	(Optional) Packets can be filtered by precedence level (as specified by a number from 0 to 7) or by the following names: • **routine** —Match packets with routine precedence (0) • **priority** —Match packets with priority precedence (1) • **immediate** —Match packets with immediate precedence (2) • **flash** —Match packets with flash precedence (3) • **flash-override** —Match packets with flash override precedence (4) • **critical** —Match packets with critical precedence (5) • **internet** —Match packets with internetwork control precedence (6) • **network** —Match packets with network control precedence (7)

dscp *dscp*	(Optional) Differentiated services code point (DSCP) provides quality of service control. The values for *dscp* are as follows: • **0**–**63**–Differentiated services codepoint value • **af11**—Match packets with AF11 dscp (001010) • **af12**—Match packets with AF12 dscp (001100) • **af13**—Match packets with AF13 dscp (001110) • **af21**—Match packets with AF21 dscp (010010) • **af22**—Match packets with AF22 dscp (010100) • **af23**—Match packets with AF23 dscp (010110) • **af31**—Match packets with AF31 dscp (011010) • **af32**—Match packets with AF32 dscp (011100) • **af33**—Match packets with AF33 dscp (011110) • **af41**—Match packets with AF41 dscp (100010) • **af42**—Match packets with AF42 dscp (100100) • **af43**—Match packets with AF43 dscp (100110) • **cs1**—Match packets with CS1 (precedence 1) dscp (001000) • **cs2**—Match packets with CS2 (precedence 2) dscp (010000) • **cs3**—Match packets with CS3 (precedence 3) dscp (011000) • **cs4**—Match packets with CS4 (precedence 4) dscp (100000) • **cs5**—Match packets with CS5 (precedence 5) dscp (101000) • **cs6**—Match packets with CS6 (precedence 6) dscp (110000) • **cs7**—Match packets with CS7 (precedence 7) dscp (111000) • **default**—Default DSCP (000000) • **ef**—Match packets with EF dscp (101110)
fragments	(Optional) Causes the software to examine fragments of IPv4 packets when applying this access list entry. When this keyword is specified, fragments are subject to the access list entry.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) **Note** ACL logging is supported only in ingress direction for both IPv4 and IPv6. The message includes the access list number, whether the packet was permitted or denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches a flow, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.
log-input	(Optional) Provides the same function as the **log** keyword, except that the log-message also includes the input interface.
icmp-off	(Optional) Turns off ICMP generation for denied packets.
icmp-type	(Optional) ICMP message type for filtering ICMP packets. Range is from 0 to 255.

icmp-code	(Optional) ICMP message code for filtering ICMP packets. Range is from 0 to 255.
igmp-type	(Optional) IGMP message type (0 to 15) or message name for filtering IGMP packets, as follows: • dvmrp • host-query • host-report • mtrace • mtrace-response • pim • precedence • trace • v2-leave • v2-report • v3-report
operator	(Optional) Operator is used to compare source or destination ports. Possible operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source* and *source-wildcard* values, it must match the source port. If the operator is positioned after the *destination* and *destination-wildcard* values, it must match the destination port. The **range** operator requires two port numbers. All other operators require one port number.
port	Decimal number of a TCP or UDP port. A port number is a number from 0 to 65535. TCP ports can be used only when filtering TCP. UDP ports can be used only when filtering UDP.
protocol-port	Name of a TCP or UDP port. TCP and UDP port names are listed in the “Usage Guidelines” section. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
established	(Optional) For the TCP protocol only: Indicates an established connection.
match-any	(Optional) For the TCP protocol only: Filters on any combination of TCP flags.
match-all	(Optional) For the TCP protocol only: Filters on all TCP flags.
+ | -	(Required) For the TCP protocol **match-any** , **match-all** : Prefix *flag-name* with **+** or **-** . Use the + *flag-name* argument to match packets with the TCP flag set. Use the - *flag-name* argument to match packets when the TCP flag is not set.
flag-name	(Optional) For the TCP protocol **match-any** , **match-all** . Flag names are: **ack** , **fin** , **psh** , **rst** , **syn** , **urg**.

Command Default

ICMP message generation is enabled by default.

Command Modes

IPv4 access list configuration

Command History

Release	Modification
Release 7.0.12	This command was introduced.
Release 7.8.1	**log-input** keyword was introduced.
Release 7.5.4	**bitmask** keyword was introduced.

Usage Guidelines

Use the **deny** command following the **ipv4 access-list** command to specify conditions under which a packet cannot pass the access list.

By default, the first statement in an access list is number 10, and the subsequent statements are incremented by 10.

You can add **permit**, **deny**, or **remark** statements to an existing access list without retyping the entire list. To add a new statement anywhere other than at the end of the list, create a new statement with an appropriate entry number that falls between two existing entry numbers to indicate where it belongs.

The following is a list of precedence names:

- critical
- flash
- flash-override
- immediate
- internet
- network
- priority
- routine

The following is a list of ICMP message type names:

- administratively-prohibited
- alternate-address
- conversion-error
- dod-host-prohibited
- dod-net-prohibited
- echo
- echo-reply
- general-parameter-problem
- host-isolated
- host-precedence-unreachable
- host-redirect
- host-tos-redirect
- host-tos-unreachable
- host-unknown
- host-unreachable

- information-reply
- information-request
- mask-reply
- mask-request
- mobile-redirect
- net-redirect
- net-tos-redirect
- net-tos-unreachable
- net-unreachable
- network-unknown
- no-room-for-option
- option-missing
- packet-too-big
- parameter-problem
- port-unreachable
- precedence-unreachable
- protocol-unreachable
- reassembly-timeout
- redirect
- router-advertisement
- router-solicitation
- source-quench
- source-route-failed
- time-exceeded
- timestamp-reply
- timestamp-request
- traceroute
- unreachable

The following is a list of TCP port names that can be used instead of port numbers. Refer to the current *Assigned Numbers* RFC to find a reference to these protocols. You can find port numbers corresponding to these protocols by typing a ? in the place of a port number.

- bgp
- chargen
- cmd
- daytime
- discard
- domain
- echo
- exec
- finger
- ftp
- ftp-data
- gopher
- hostname
- ident

- irc
- klogin
- kshell
- login
- lpd
- nntp
- pim-auto-rp
- pop2
- pop3
- smtp
- sunrpc
- tacacs
- talk
- telnet
- time
- uucp
- whois
- www

The following UDP port names can be used instead of port numbers. Refer to the current *Assigned Numbers* RFC to find a reference to these protocols. You can find port numbers corresponding to these protocols by typing a ? in the place of a port number.

- biff
- bootpc
- bootps
- discard
- dnsix
- domain
- echo
- isakmp
- mobile-ip
- nameserver
- netbios-dgm
- netbios-ns
- netbios-ss
- ntp
- pim-auto-rp
- rip
- snmp
- snmptrap
- sunrpc
- syslog
- tacacs
- talk
- tftp
- time

- who
- xdmcp

Use the following flags in conjunction with the **match-any** and **match-all** keywords and the + and - signs to select the flags to display:

- ack
- fin
- psh
- rst
- syn

For example, **match-all** + *ack* + *syn* displays TCP packets with both the ack *and* syn flags set, or **match-any** + *ack* - *syn* displays the TCP packets with the ack set *or* the syn not set.

Note If any ACE in an ACL contains ABF clause, this ACL cannot be applied at any non-zero compression level.

Task ID

Task ID	Operations
ipv4	read, write
acl	read, write

Examples

This example shows how to set a deny condition for an access list named Internet filter:

```
Router(config)# ipv4 access-list Internetfilter
Router(config-ipv4-acl)# 10 deny 192.168.34.0 0.0.0.255
Router(config-ipv4-acl)# 20 deny 172.16.0.0 0.0.255.255
Router(config-ipv4-acl)# 25 deny tcp host 172.16.0.0 gt bgp host 192.168.202.203 range 1300
 1400
Router(config-ipv4-acl)# permit 10.0.0.0 0.255.255.255
```

This example shows how you can configure DSCP bitmask on ingress ERSPAN.

```
Router# config
Router(config)# ipv4 access-list acl1
Router(config-ipv4-acl)# 10 permit ipv4 host 192.0.2.1 any dscp af22 bitmask 0x3f
Router(config-ipv4-acl)# commit
Router(config-ipv4-acl)# exit
Router(config)# interface HundredGigE0/0/0/6
Router(config-if)# ipv4 address 192.0.2.51 255.255.255.0
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv4 acl1
Router(config-if)# commit
```

deny (IPv6)

To set deny conditions for an IPv6 access list, use the **deny** command in IPv6 access list configuration mode. To remove the deny conditions, use the **no** form of this command.

[sequence-number] **deny** *protocol* { *source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/ prefix-length* } [*operator* { *port* | *protocol-port* }] [**dscp** *value* [**bitmask** *value*]] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log | log-input**] [**ttl** *ttl value* [*value1* . . . *value2*]] **icmp-off**]
no *sequence-number*

Internet Control Message Protocol (ICMP)
[sequence-number] **deny icmp** { *source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/ prefix-length* } { *destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/ prefix-length* } [*icmp-type*] [*icmp-code*] [**dscp** *value*] [`routing`] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**log**] [**icmp-off**]

Transmission Control Protocol (TCP)
*[sequence-number]***deny tcp** {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*operator* {*port* | *protocol* | *port*}] [**dscp***value*] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**established**] {**match-any | match-all | + | -**} [*flag-name*] [**log**] [**icmp-off**]

User Datagram Protocol (UDP)
*[sequence-number]***deny tcp** {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/ prefix-length*} [*operator* {*port* | *protocol* | *port*}] [**dscp***value*] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**established**] [*flag-name*] [**log**] [**icmp-off**]

Syntax Description

sequence-number	(Optional) Number of the **deny** statement in the access list. This number determines the order of the statements in the access list. Range is 1 to 2147483644. (By default, the first statement is number 10, and the subsequent statements are incremented by 10.)
protocol	Name or number of an Internet protocol. It can be one of the keywords **ahp** , **esp** , **gre**, **icmp** , **igmp**, **igrp**, **ipinip**, **ipv6** , **nos**, **ospf**, **pcp** , **tcp** , or **udp** , or an integer in the range from 0 to 255 representing an IPv6 protocol number.
source-ipv6-prefix / *prefix-length*	The source IPv6 network or class of networks about which to set deny conditions. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
any	An abbreviation for the IPv6 prefix ::/0.
host *source-ipv6-address*	Source IPv6 host address about which to set deny conditions. This *source-ipv6-address* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

ipv6-wildcard-mask	IPv6 wildcard mask. The IPv6 wildcard mask can take any IPv6 address value which is used instead of prefix length.
operator {*port* | *protocol-port*}	(Optional) Operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source-ipv6-prefix / prefix-length* argument, it must match the source port. If the operator is positioned after the *destination-ipv6-prefix / prefix-length* argument, it must match the destination port. The **range** operator requires two port numbers. All other operators require one port number. The *port* argument is the decimal number of a TCP or UDP port. Range is 0 to 65535. The *protocol-port* argument is the name of a TCP or UDP port. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
destination-ipv6-prefix / prefix-length	Destination IPv6 network or class of networks about which to set deny conditions. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
host *destination-ipv6-address*	Destination IPv6 host address about which to set deny conditions. This *destination-ipv6-address* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
dscp *value*	(Optional) Matches a differentiated services code point DSCP value against the traffic class value in the Traffic Class field of each IPv6 packet header. Range is 0 to 63.
routing	(Optional) Matches source-routed packets against the routing extension header within each IPv6 packet header.
hop-by-hop	(Optional) Supports Jumbo-grams. With the Router Alert option, it is an integral part in the operation of Multicast Listener Discovery (MLD). Router Alert [3] is an integral part in the operations of IPv6 Multicast through MLD and RSVP for IPv6.
authen	(Optional) Matches if the IPv6 egress authentication header is present.
destopts	(Optional) Matches if the IPv6 egress destination options header is present.
fragments	(Optional) Matches noninitial fragmented packets where the fragment extension header contains a nonzero fragment offset. The **fragments** keyword is an option only if the *operator* [*port-number*] arguments are not specified.

log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) **Note** ACL logging is supported only in ingress direction for both IPv4 and IPv6. The message includes the access list name and sequence number, whether the packet was denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets denied in the prior 5-minute interval.
log-input	(Optional) Provides the same function as the **log** keyword, except that the log-message also includes the input interface.
ttl	(Optional) Turns on matching against time-to-life (TTL) value. For IPv6 packets, **ttl** is also referred to as hop limit.
ttl value [*value1 ... value2*]	(Optional) TTL value used for filtering. Range is 1 to 255. If only *value* is specified, the match is against this value. If both *value1* and *value2* are specified, the packet TTL is matched against the range of TTLs between *value1* and *value2* .
operator	(Optional) Operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).
icmp-off	(Optional) Turns off ICMP generation for denied packets.
icmp-type	(Optional) ICMP message type for filtering ICMP packets. ICMP packets can be filtered by ICMP message type. Range is 0 to 255.
icmp-code	(Optional) ICMP message code for filtering ICMP packets. ICMP packets that are filtered by ICMP message type can also be filtered by the ICMP message code. Range is 0 to 255.
established	(Optional) For the TCP protocol only: Indicates an established connection.
match-any	(Optional) For the TCP protocol only: Filters on any combination of TCP flags.
match-all	(Optional) For the TCP protocol only: Filters on all TCP flags.
+ | -	(Required) For the TCP protocol **match-any** , **match-all** : Prefix *flag-name* with **+** or **-** . Use the + *flag-name* argument to match packets with the TCP flag set. Use the - *flag-name* argument to match packets when the TCP flag is not set.
flag-name	(Optional) For the TCP protocol **match-any** , **match-all** . Flag names are: **ack**, **fin**, **psh**, **rst**, **syn**, **urg**.

Command Default

ICMP message generation is enabled by default.

Command Modes

IPv6 access list configuration

Command History

Release	Modification
Release 7.0.12	This command was introduced.
Release 7.2.1	Ingress IPv6 TCP flags are supported.
Release 7.3.15	Egress IPv6 TCP flags are supported.
Release 7.8.1	**log-input** keyword was introduced.
Release 7.8.1	**ttl** keyword was introduced.
Release 7.5.4	**bitmask** keyword was introduced.
Release 7.10.1	IPv6 AHP and ESP headers are supported.

Usage Guidelines

The **deny** (IPv6) command is similar to the **deny** (IPv4) command, except that it is IPv6-specific.

Use the **deny** (IPv6) command following the **ipv6 access-list** command to define the conditions under which a packet passes the access list.

Note If any ACE in an ACL contains ABF clause, this ACL cannot be applied at any non-zero compression level.

Specifying **ipv6** for the *protocol* argument matches against the IPv6 header of the packet.

By default, the first statement in an access list is number 10, and the subsequent statements are numbered in increments of 10.

You can add permit, deny, or remark statements to an existing access list without retyping the entire list. To add a new statement anywhere other than at the end of the list, create a new statement with an appropriate entry number that falls between two existing entry numbers to indicate where it belongs.

Both the *source-ipv6-prefix/prefix-length* and *destination-ipv6-prefix/prefix-length* arguments are used for traffic filtering (the source prefix filters traffic based upon the traffic source; the destination prefix filters traffic based upon the traffic destination).

Note IPv6 prefix lists, not access lists, should be used for filtering routing protocol prefixes.

The **fragments** keyword is an option only if the *operator* [*port* | *protocol-port*] arguments are not specified.

Task ID

Task ID	Operations
acl	read, write

Examples

The following example shows how to configure the IPv6 access list named toCISCO and apply the access list to the traffic entering the HundredGigE interface 0/2/0/2. Specifically, the deny entry in the list keeps all packets that have a destination TCP port number greater than 5000 from entering the HundredGigE interface 0/2/0/2. The permit entry in the list permits all ICMP packets to enter the HundredGigE interface 0/2/0/2.

```
Router(config)# ipv6 access-list toCISCO
Router(config-ipv6-acl)# deny tcp any any gt 5000
Router(config-ipv6-acl)# permit icmp any any
Router(config)#  interface HundredGigE 0/2/0/2
Router(config-if)# ipv6 access-group tOCISCO ingress
```

The following example shows how to configure the IPv6 access list named toCISCO and apply the access list to the traffic entering theHundredGigE interface 0/2/0/2. Specifically, the deny entry in the list keeps all packets that have a hop-by-hop optional field from entering the HundredGigE interface 0/2/0/2.

```
Router(config)# ipv6 access-list toCISCO
Router(config-ipv6-acl)# deny ipv6 any any hop-by-hop
Router(config)#  interface HundredGigE 0/2/0/2
Router(config-if)# ipv6 access-group tOCISCO ingress
```

The following example shows how you can configure DSCP bitmask on ingress ERSPAN.

```
Router# config
Router(config)# ipv6 access-list acl1
Router(config-ipv6-acl)# 10 permit ipv6 host 2001:DB8::2/32 any dscp 33 bitmask 0x3f
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit
Router(config)# interface HundredGigE 0/0/10/3
Router(config-if)# ipv6 address 2001:DB8::1/32
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv6 acl1
Router(config-if)# commit
```

The following example shows how you can configure AHP and ESP headers on an ACLs.

```
Router(config)# #ipv6 access-list ipv6_umpp_access_list
Router(config-ipv6-acl)# 12 deny ahp any any
Router(config-ipv6-acl)# ipv6 access-list ipv6_umpp_access_list
Router(config-ipv6-acl)# 14 deny esp any any
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit
```

dont-fragment

To configure an access list to match on the **dont-fragment** flag.

fragment-type dont-fragment {**capture** | **counter** | **first-fragment** | **is-fragment** | **last-fragment** | **log** | **log-input** | **set** | **udf** | **nexthop1** }

Syntax Description

capture	ACL matches on the **dont-fragment** flag, and captures the matched packet.
counter	ACL matches on the **dont-fragment** flag, and displays the counter for the matches.
first-fragment	ACL matches on the **dont-fragment** flag, and then matches on the **first-fragment** flag.
is-fragment	ACL matches on the **dont-fragment** flag, and then matches on the **is-fragment** flag.
last-fragment	ACL matches on the **dont-fragment** flag, and then matches on the **last-fragment** flag.
log	ACL matches on the **dont-fragment** flag and logs the matches.
log-input	ACL matches on the **dont-fragment** flag and logs the matches, incuding on the input interface.
set	ACL matches on the **dont-fragment** flag and sets a particular action on the matches.
udf	ACL matches on the **dont-fragment** flag, and sets the user-defined fields for the matches.
nexthop1	ACL matches on the **dont-fragment** flag, and then matches on the **nexthop1** flag.

Command Default

None

Command Modes

ACL configuration mode

Command History

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

This command is supported only for IPv4 ACLs.

Example

Use the following sample configuration to match on the **dont-fragment** flag.

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the dont-fragment flag (indicates a non-fragmented packet)
 and forward the packet to the default (pre-configured) next hop  */
Router(config-ipv4-acl)# 20 permit tcp any any fragment-type dont-fragment nexthop1 ipv4
```

```
192.0.2.1
Router(config-ipv4-acl)# commit
```

first-fragment

To configure an ACL to match on the **first-fragment** flag.

fragment-type first-fragment {**capture** | **counter** | **log** | **log-input** | **set** | **udf** | **<none>**}

Syntax Description

capture	ACL matches on the **first-fragment** flag, and captures the matched packet.
counter	ACL matches on the **first-fragment** flag, and displays the counter for the matches.
log	ACL matches on the **first-fragment** flag and logs the matches.
log-input	ACL matches on the **first-fragment** flag and logs the matches, incuding on the input interface.
set	ACL matches on the **first-fragment** flag and sets a particular action on the matches.
udf	ACL matches on the **first-fragment** flag, and sets the user-defined fields for the matches.
nexthop1	ACL matches on the **first-fragment** flag, and then matches on the **nexthop1** flag.

Command Default

None

Command Modes

ACL configuration mode.

Command History

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

This command is supported only for IPv4 ACLs.

Example

Use the following sample configuration to match on the **first-fragment** flag.

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the first-fragment flag (indicates the first fragment of a
 fragmented packet)
 and forward the packet to a next hop of 20.20.20.1  */
Router(config-ipv4-acl)# 40 permit ospf any any fragment-type first-fragment nexthop1 ipv4
 192.0.2.1
Router(config-ipv4-acl)# commit
```

fragment-offset

To enable packet filtering at an ingress or egress interface by specifying fragment-offset as a match condition in an IPv4 or IPv6 ACL, use the **fragment-offset** option in **permit** or **deny** command in IPv4 or IPv6 access-list configuration mode. To disable this feature, use the **no** form of this command.

fragment-offset {**eq** *value* | **gt** *value* | **lt** *value* | **neq** *value* | **range** *lower-limit upper-limit*}

Syntax Description

fragment-offset eq *value*	Filters packets that have a fragment offset equal to the specified limit.
fragment-offset gt *value*	Filters packets that have a fragment offset greater than the specified limit.
fragment-offset lt *value*	Filters packets that have a fragment offset less than the specified limit.
fragment-offset neq *value*	Filters packets that have a fragment offset that does not match the specified limit.
fragment-offset range *lower-limit upper-limit*	Filters packets that have a fragment offset within the specified range.

Command Default

None

Command Modes

IPv4 or IPv6 Access List Configuration mode

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Example

This example shows how to configure an IPv4 access list to filter packets by the fragment-offset condition:

```
Router# config
Router(config)# ipv4 access-list fragment-offset-acl
Router(config-ipv4-acl)# 10 permit ipv4 any any fragment-offset range 300 400
```

fragment-type

To configure an access list to match on the type of fragment.

fragment-type {**dont-fragment** | **first-fragment** | **is-fragment** | **last-fragment**}

Syntax Description

dont-fragment	ACL matches on the **dont-fragment** flag
first-fragment	ACL matches on the **first-fragment** flag
is-fragment	ACL matches on the **is-fragment** flag
last-fragment	ACL matches on the **last-fragment** flag

Command Default

None

Command Modes

ACL configuration mode

Command History

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

This command is supported only for IPv4 access lists.

Example

Use the following sample configuration to configure an ACL to match on the type of fragment..

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the dont-fragment flag (indicates a non-fragmented packet)
 and forward the packet to the default (pre-configured) next hop  */
Router(config-ipv4-acl)# 20 permit tcp any any fragment-type dont-fragment default

/* Configure an ACE to match on the is-fragment flag (indicates a fragmented packet)
 and forward the packet to a next hop of 10.10.10.1  */
Router(config-ipv4-acl)# 30 permit udp any any fragment-type is-fragment nexthop1 ipv4
10.10.10.1

/* Configure an ACE to match on the first-fragment flag (indicates the first fragment of a
 fragmented packet)
 and forward the packet to a next hop of 20.20.20.1  */
Router(config-ipv4-acl)# 40 permit ospf any any fragment-type first-fragment nexthop1 ipv4
 20.20.20.1


/* Configure an ACE to match on the last-fragment flag (indicates the last fragment of a
fragmented packet)
 and forward the packet to a next hop of 30.30.30.1  */
```

```
Router(config-ipv4-acl)# 50 permit icmp any any fragment-type last-fragment nexthop1 ipv4
30.30.30.1
Router(config-ipv4-acl)# commit
```

hw-module profile stats acl-permit

To obtain statistics of the packet count of the routing traffic that an ACL permits, use the **hw-module profile stats acl-permit** command in XR Config mode. To disable the tracking of permitted packet count, use the **no** form of this command.

hw-module profile stats acl-permit
no hw-module profile stats acl-permit

Syntax Description

This command has no keywords or arguments.

Command Default

If you do not configure the **hw-module profile stats acl-permit** command, you cannot enable the statistics for the routing traffic that an ACL permits.

Command Mode

XR Config

Command History

Release	Modification
Release 7.3.2	Supports logging of permit statistics for ACL-based forwarding (ABF).
Release 7.2.12	This command was introduced.

Usage Guidelines

- The permit statistics of the routing traffic that an ACL allows are available only after you execute the **hw-module profile stats acl-permit** command and based on the requirement, reboot the line cards or the router.

Task ID	Operations
config-services	read, write
root-lr	read, write

Examples

The following example shows you how to configure the **acl-permit** command:

```
Router# configure
Router(config)# hw-module profile stats acl-permit
Fri Aug  7 05:52:58.052 UTC
In order to activate/deactivate this stats profile, you must manually reload the chassis/all
 line cards
Router(config)# commit
Fri Aug 7 05:55:50.103 UTC
```

```
LC/0/4/CPU0:Aug 7 05:55:50.218 UTC: fia_driver[245]:
%FABRIC-FIA_DRVR-4-STATS_HW_PROFILE_MISMATCH : Mismatch found, reload LC to activate the
new stats profile
Router(config)#
```

ipv4 access-group

To control access to an interface, use the **ipv4 access-group** command in interface configuration mode. To remove the specified access group, use the **no** form of this command.

ipv4 access-group *access-list-name* { **ingress** | **egress** } [**compress level** *compression-level*]

Syntax Description

access-list-name	Name of an IPv4 access list as specified by an **ipv6 access-list** command.
ingress	Filters on inbound packets.
egress	Filters on outbound packets.
compress level *compression-level*	Configures compression level for interface ACLs. Compression level values range from zero and five.

Command Default

The interface does not have an IPv4 access list applied to it.

Command Modes

Interface configuration

Command History

Release	Modification
Release 7.0.12	This command was introduced.
Release 7.3.1	Compression level can be configured

Usage Guidelines

Use the **ipv4 access-group** command to control access to an interface. To remove the specified access group, use the **no** form of the command. Use the *access-list-name* argument to specify a particular IPv4 access list.

Filtering of MPLS packets through interface ACL is not supported.

If the access list permits the addresses, the software continues to process the packet. If the access list denies the address, the software discards the packet and returns an Internet Control Message Protocol (ICMP) host unreachable message.

If the specified access list does not exist, all packets are passed.

By default, the unique or per-interface ACL statistics are disabled.

Task ID

Task ID	Operations
acl	read, write
network	read, write

Examples

The following example shows how to apply filters on packets from HundredGigE interface 0/2/0/2:

```
Router(config)# interface HundredGigE 0/2/0/2
Router(config-if)# ipv4 access-group p-ingress-filter ingress
```

The following example shows how to apply compress level 2 on ingress traffic:

```
Router(config)# interface HundredGigE 0/2/0/0
Router(config-if)# ipv4 access-group p-ingress-filter ingress compress level 2
```

This example shows how to apply compression level 2 on egress traffic for an IPv4 Hybrid ACL, where you've already created a network object group and attached an ACL(network-object-acl) to it:

```
Router# configure
Router(config)# interface HundredGigE 0/0/10/3
Router(config-if)# ipv4 address 1.1.1.1/24
Router(config-if)# no shut
Router(config-if)# ipv4 access-group network-object-acl egress compress level 2
Router(config-if)# commit
Router(config-if)# exit
```

ipv4 access-list

To define an IPv4 access list by name, use the **ipv4 access-list** command in XR Config mode. To remove all entries in an IPv4 access list, use the **no** form of this command.

ipv4 access-list *name*
no ipv4 access-list *name*

Syntax Description

name	Name of the access list. Names cannot contain a space or quotation marks.

Command Default

No IPv4 access list is defined.

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

Use the **ipv4 access-list** command to configure an IPv4 access list. This command places the router in access list configuration mode, in which the denied or permitted access conditions must be defined with the **deny** or **permit** command.

Use the **ipv4 access-group** command to apply the access list to an interface.

Task ID

Task ID	Operations
acl	read, write

Examples

This example shows how to define a standard access list named Internetfilter:

```
Router(config)# ipv4 access-list Internetfilter
Router(config-ipv4-acl)# 10 permit 192.168.34.0 0.0.0.255
Router(config-ipv4-acl)# 20 permit 172.16.0.0 0.0.255.255
Router(config-ipv4-acl)# 30 permit 10.0.0.0 0.255.255.255
Router(config-ipv4-acl)# 39 remark Block BGP traffic from 172.16 net.
Router(config-ipv4-acl)# 40 deny tcp host 172.16.0.0 eq bgp host 192.168.202.203 range 1300
 1400
```

ipv4 access-list log-update rate

To specify the rate at which IPv4 access lists are logged, use the **ipv4 access-list log-update rate** command in XR Config mode. To return the update rate to the default setting, use the **no** form of this command.

ipv4 access-list log-update rate *rate-number*
no ipv4 access-list log-update rate *rate-number*

Syntax Description

rate-number	Rate at which IPv4 access hit logs are generated per second on the router. Range is 1 to 1000.

Command Default

Default is 1.

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

The *rate-number* argument applies to all the IPv4 access-lists configured on the interfaces. That is, at any given time there can be between 1 and 1000 log entries for the system.

Task ID

Task ID	Operations
ipv4	read, write
acl	read, write

Examples

The following example shows how to configure a IPv4 access hit logging rate for the system:

```
Router(config)# ipv4 access-list log-update rate 10
```

ipv4 access-list log-update threshold

To specify the number of updates that are logged for IPv4 access lists, use the **ipv4 access-list log-update threshold** command in XR Config mode. To return the number of logged updates to the default setting, use the **no** form of this command.

ipv4 access-list log-update threshold *update-number*
no ipv4 access-list log-update threshold *update-number*

Syntax Description

update-number	Number of updates that are logged for every IPv4 access list configured on the router. Range is 0 to 2147483647.

Command Default

For IPv4 access lists, 2147483647 updates are logged.

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

IPv4 access list updates are logged at 5-minute intervals, following the first logged update. Configuring a lower number of updates (a number lower than the default) is useful when more frequent update logging is desired.

Task ID

Task ID	Operations
basic-services	read, write
acl	read, write

Examples

This example shows how to configure a log threshold of ten updates for every IPv4 access list configured on the router:

```
Router(config)# ipv4 access-list log-update threshold 10
```

ipv6 access-group

To control access to an interface, use the **ipv6 access-group** command in interface configuration mode. To remove the specified access group, use the **no** form of this command.

ipv6 access-group *access-list-name* { **ingress** | **egress** } [**compress level** *compression-level*]

Syntax Description

access-list-name	Name of an IPv6 access list as specified by an **ipv6 access-list** command.
ingress	Filters on inbound packets.
compress level *compression-level*	Configures compression level for interface ACLs. Compression level values range from zero and five.

Command Default

The interface does not have an IPv6 access list applied to it.

Command Modes

Interface configuration

Command History

Release	Modification
Release 7.0.12	This command was introduced.
Release 7.3.1	Compression level can be configured

Usage Guidelines

Use compression level two to create Hybrid ACLs with an ACE that uses IPv6 extension headers to filter ingress and egress IPv6 packets.

Task ID

Task ID	Operations
acl	read, write
ipv6	read, write

Examples

This example shows how to apply filters on packets from HundredGigE interface 0/2/0/2:

```
Router(config)# interface HundredGigE 0/2/0/2
Router(config-if)# ipv6 access-group p-in-filter ingress
```

This example shows how to create an ingress IPv6 Hybrid ACL with compression level 2 based on extensions headers:

```
Router# configure
Router(config)# ipv6 access-list ACL-EXT-HEADER
Router(config-ipv6-acl)# 10 deny ipv6 any any routing
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit
Router(config)# interface hundredGigE 0/4/0/36
Router(config-if)# ipv6 access-group ACL-EXT-HEADER ingress compress level 2
Router(config-if)# commit
```

This example shows how to create an egress IPv6 Hybrid ACL with compression level 2 based on extensions headers:

```
Router# configure
Router(config)# ipv6 access-list ACL-EGRESS
Router(config-ipv6-acl)# 10 deny ipv6 any any routing
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit
Router(config)# interface hundredGigE 0/4/0/13
Router(config-if)# ipv6 access-group ACL-EGRESS egress compress level 2
Router(config-if)# commit
```

ipv6 access-list

To define an IPv6 access list and to place the router in IPv6 access list configuration mode, use the **ipv6 access-list** command in interface configuration mode. To remove the access list, use the **no** form of this command.

ipv6 access-list *name*
no ipv6 access-list *name*

Syntax Description

name	Name of the IPv6 access list. Names cannot contain a space or quotation mark, or begin with a numeric.

Command Default

No IPv6 access list is defined.

Command Modes

Interface configuration

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

The **ipv6 access-list** command is similar to the **ipv4 access-list** command, except that it is IPv6-specific.

The IPv6 access lists are used for traffic filtering based on source and destination addresses, IPv6 option headers, and optional, upper-layer protocol type information for finer granularity of control. IPv6 access lists are defined by using the **ipv6 access-list** command in XR Config mode mode and their permit and deny conditions are set by using the **deny** and **permit** commands in IPv6 access list configuration mode. Configuring the **ipv6 access-list** command places the router in IPv6 access list configuration mode—the router prompt changes to router (config-ipv6-acl)#. From IPv6 access list configuration mode, permit and deny conditions can be set for the defined IPv6 access list.

See the “Examples” section for an example of a translated IPv6 access control list (ACL) configuration.

Note No more than one IPv6 access list can be applied to an interface per direction.

Note Every IPv6 access list has an implicit **deny ipv6 any any** statement as its last match condition. An IPv6 access list must contain at least one entry for the implicit **deny ipv6 any any** statement to take effect.

Note IPv6 prefix lists, not access lists, should be used for filtering routing protocol prefixes.

Use the **ipv6 access-group** interface configuration command with the *access-list-name* argument to apply an IPv6 access list to an IPv6 interface.

Note An IPv6 access list applied to an interface with the **ipv6 access-group** command filters traffic that is forwarded, not originated, by the router.

Note Every IPv6 ACL has implicit **permit icmp any any nd-na**, **permit icmp any any nd-ns**, and **deny ipv6 any any** statements as its last match conditions. (The former two match conditions allow for ICMPv6 neighbor discovery.) An IPv6 ACL must contain at least one entry for the implicit **deny ipv6 any any** statement to take effect.**permit icmp any any nd-na permit icmp any any nd-ns deny ipv6 any any deny ipv6 any any**.

The IPv6 neighbor discovery process makes use of the IPv6 network layer service; therefore, by default, IPv6 ACLs implicitly allow IPv6 neighbor discovery packets to be sent and received on an interface. In IPv4, the Address Resolution Protocol (ARP), which is equivalent to the IPv6 neighbor discovery process, makes use of a separate data link layer protocol; therefore, by default, IPv4 ACLs implicitly allow ARP packets to be sent and received on an interface.

Task ID

Task ID	Operations
acl	read, write
ipv6	read, write

Examples

This example shows how to configure the IPv6 access list named list2 and applies the ACL to traffic on interface HundredGigE 0/2/0/2. Specifically, the first ACL entry keeps all packets from the network fec0:0:0:2::/64 (packets that have the site-local prefix fec0:0:0:2 as the first 64 bits of their source IPv6 address) from exiting out of interface HundredGigE 0/2/0/2. The second entry in the ACL permits all other traffic to exit out of interface HundredGigE 0/2/0/2. The second entry is necessary because an implicit deny all condition is at the end of each IPv6 ACL.

```
Router(config)# ipv6 access-list list2
Router(config-ipv6-acl)# 10 deny fec0:0:0:2::/64 any
Router(config-ipv6-acl)# 20 permit any any

Router# show ipv6 access-lists list2

ipv6 access-list list2
  10 deny ipv6 fec0:0:0:2::/64 any
  20 permit ipv6 any any

Router(config)# interface HundredGigE 0/2/0/2
```

Note IPv6 is automatically configured as the protocol type in **permit any any** and **deny any any** statements that are translated from XR Config mode mode to IPv6 access list configuration mode.

Note An IPv6 router does not forward to another network an IPv6 packet that has a link-local address as either its source or destination address (and the source interface for the packet is different from the destination interface for the packet).

ipv6 access-list log-update rate

To specify the rate at which IPv6 access lists are logged, use the **ipv6 access-list log-update rate** command in XR Config mode. To return the update rate to the default setting, use the **no** form of this command.

ipv6 access-list log-update rate *rate-number*
no ipv6 access-list log-update rate *rate-number*

Syntax Description

rate-number	Rate at which IPv6 access hit logs are generated per second on the router. Range is 1 to 1000.

Command Default

Default is 1.

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

The *rate-number* argument applies to all the IPv6 access-lists configured on the interfaces. That is, at any given time there can be between 1 and 1000 log entries for the system.

Task ID

Task ID	Operations
ipv6	read, write
acl	read, write

Examples

This example shows how to configure a IPv6 access hit logging rate for the system:

```
Router(config)# ipv6 access-list log-update rate 10
```

ipv6 access-list log-update threshold

To specify the number of updates that are logged for IPv6 access lists (ACLs), use the **ipv6 access-list log-update threshold** command in XR Config mode. To return the number of logged updates to the default setting, use the **no** form of this command.

ipv6 access-list log-update threshold *update-number*
no ipv6 access-list log-update threshold *update-number*

Syntax Description

update-number	Number of updates that are logged for every IPv6 access list configured on the router. Range is 0 to 2147483647.

Command Default

For IPv6 access lists, 350000 updates are logged.

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

The **ipv6 access-list log-update threshold** command is similar to the **ipv4 access-list log-update threshold** command, except that it is IPv6-specific.

IPv6 access list updates are logged at 5-minute intervals, following the first logged update. Configuring a lower number of updates (a number lower than the default) is useful when more frequent update logging is desired.

Task ID

Task ID	Operations
acl	read, write
ipv6	read, write

Examples

This example shows how to configure a log threshold of ten updates for every IPv6 access list configured on the router:

```
Router(config)# ipv6 access-list log-update threshold 10
```

is-fragment

To configure an ACL to match on the **is-fragment** flag.

fragment-type is-fragment {**capture** | **counter** | **log** | **log-input** | **set** | **udf** | **nexthop1** }

Syntax Description

capture	ACL matches on the **is-fragment** flag, and captures the matched packet.
counter	ACL matches on the **is-fragment** flag, and displays the counter for the matches.
log	ACL matches on the **is-fragment** flag and logs the matches.
log-input	ACL matches on the **is-fragment** flag and logs the matches, incuding on the input interface.
set	ACL matches on the **is-fragment** flag and sets a particular action on the matches.
udf	ACL matches on the **is-fragment** flag, and sets the user-defined fields for the matches.
nexthop1	ACL matches on the **is-fragment** flag, and then matches on the **nexthop1** flag.

Command Default

None

Command Modes

ACL configuration mode.

Command History

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

This command is supported only for IPv4 ACLs.

Example

Use the following sample configuration to match on the **is-fragment** flag.

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the is-fragment flag (indicates a fragmented packet)
 and forward the packet to a next hop of 10.10.10.1  */
Router(config-ipv4-acl)# 30 permit udp any any fragment-type is-fragment nexthop1 ipv4
192.0.2.1
Router(config-ipv4-acl)# commit
```

last-fragment

To configure an access list to match on the **last-fragment** flag.

fragment-type last-fragment {**capture** | **counter** | **log** | **log-input** | **set** | **udf** | **nexthop1** }

Syntax Description

capture	ACL matches on the **last-fragment** flag, and captures the matched packet.
counter	ACL matches on the **last-fragment** flag, and displays the counter for the matches.
log	ACL matches on the **last-fragment** flag and logs the matches.
log-input	ACL matches on the **last-fragment** flag and logs the matches, incuding on the input interface.
set	ACL matches on the **dont-fragment** flag and sets a particular action on the matches.
udf	ACL matches on the **last-fragment** flag, and sets the user-defined fields for the matches.
nexthop1	ACL matches on the **last-fragment** flag, and then matches on the **nexthop1** flag.

Command Default

None

Command Modes

ACL configuration mode.

Command History

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

This command is supported only for IPv4 ACLs.

Example

Use the following sample configuration to match on the **last-fragment** flag.

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the last-fragment flag (indicates the last fragment of a
fragmented packet)
 and forward the packet to a next hop of 30.30.30.1  */
Router(config-ipv4-acl)# 50 permit icmp any any fragment-type last-fragment nexthop1 ipv4
192.0.2.1
Router(config-ipv4-acl)# commit
```

object-group network

To configure a network object group, and to enter the network object group configuration mode, use the **object-group network** command in the global configuration mode. To de-configure the network object group, use the **no** form of this command.

object-group network { **ipv4 | ipv6** } *object-group-name*
no object-group network { **ipv4 | ipv6** } *object-group-name*

Syntax Description

ipv4	Configures the operation state of an IPV4 network object group.
ipv6	Configures the operation state of an IPV6 network object group.
object-group-name	Name of the object-group.

Command Default

None

Command Modes

Global configuration

Command History

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

Inherited object-groups up to four levels are supported in this release.

If an ACL is applied on an interface with non-zero compression level (implying it contains no ABF ACEs), a user cannot add an ACE with object-group.

Task ID

Task ID	Operation
system	read, write

Example

This example shows how to configure a network object-group, and to enter the network object-group configuration mode:

```
Router# configure
Router(config)# object-group network ipv4 ipv4_type5_obj1
Router(config-object-group-ipv4)#
```

object-group port

To configure a port object group, and to enter the port object group configuration mode, use the **object-group port** command in the global configuration mode. To de-configure the port object group, use the **no** form of this command.

object-group port *object-group-name*
no object-group port *object-group-name*

Syntax Description

object-group-name	Name of the object-group.

Command Default

None

Command Modes

Global configuration

Command History

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

Inherited object-groups upto four levels are supported.

Note If any ACE in an ACL contains ABF clause, this ACL cannot be applied at any non-zero compression level.

Task ID

Task ID	Operation
system	read, write

Example

This example show how to configure a port object-group, and to enter the port object-group configuration mode:

```
Router# configure
Router(config)# object-group port ipv4_type5_obj1
Router(config-object-group-port)#
```

packet-length

Enables filtering of packets at an ingress/egress interface by specifying the packet length as a match condition in a IPv4/IPv6 ACL.

By using the **packet-length** condition in an ACL, IPv4 and IPv6 packets are either processed (permit statement) or dropped (deny statement).

To remove this configuration, use the **no** prefix for the command.

packet-length { **eq** *value* | **gt** *value* | **lt** *value* | **neq** *value* | **range** *lower-limit upper-limit* }

Syntax Description

packet-length eq *value*	Filters packets that have a packet length equal to the specified limit.
packet-length gt *value*	Filters packets that have a packet length greater than the specified limit.
packet-length lt *value*	Filters packets that have a packet length less than the specified limit.
packet-length neq *value*	Filters packets that have a packet length that does not match the specified limit.
packet-length range *lower-limit upper-limit*	Filters packets that have a packet length within the specified range. The IPv4/IPv6 packet length ranges from 0 to 65535.

Command Default

None

Command Modes

Access List Configuration mode

Release	Modification
Release 7.3.1	This command was introduced.

Usage Guidelines

No specific guidelines impact the use of this command.

Example

The following example shows how you can configure an IPv4 access list with the **packet-length** condition.

```
Router# config
Router(config)# ipv4 access-list pktlen-v4
Router(config-ipv4-acl)# 10 permit tcp any any packet-length eq 1482
Router(config-ipv4-acl)# 20 permit udp any any packet-length range 1400 1500
Router(config-ipv4-acl)# 30 deny ipv4 any any
```

The following example shows how you can configure an IPv6 access list with the **packet-length** condition.

```
Router# config
Router(config)# ipv6 access-list pktlen-v6
Router(config-ipv6-acl)# 10 permit tcp any any packet-length eq 1500
Router(config-ipv6-acl)# 20 permit udp any any packet-length range 1500 1600
Router(config-ipv6-acl)# 30 deny ipv6 any any
```

permit (IPv4)

To set conditions for an IPv4 access list, use the **permit** command in access list configuration mode. There are two versions of the **permit** command: **permit** (source), **permit** (destination), and **permit** (protocol). To remove a condition from an access list, use the **no** form of this command.

[*sequence-number*] **permit** *source* [*source-wildcard*] [**log | log-input**]
[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**nexthop** [*ipv4-address1*] [*ipv4-address2*] [*ipv4-address3*]] [**dscp** *dscp* [**bitmask** *value*]] [**fragments**] [**log | log-input**] [**nexthop** [**track** *track-name*] [*ipv4-address1*] [*ipv4-address2*] [*ipv4-address3*] [**ttl** *ttl value* [*value1* **. . .** *value2*]]
no *sequence-number*

Internet Control Message Protocol (ICMP)
[*sequence-number*] **permit icmp** *source source-wildcard destination destination-wildcard* [*icmp-type*] [*icmp-code*] [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**]

Transmission Control Protocol (TCP)
[sequence-number] **permit tcp** { *source-ipv4-prefix/ prefix-length* | *any* | *host source-ipv4-address ipv4-wildcard-mask/prefix-length* } [*operator* { *port* | *protocol-port* }] { *destination-ipv4-prefix/ prefix-length* | *any* | *host destination-ipv4-address ipv4-wildcard-mask/prefix-length* } [*operator* { *port* | *protocol* | *port* }] [**dscp** *value*] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**established**] { **match-any | match-all | + | -** } [*flag-name*] [**log**]

Internet Group Management Protocol (IGMP)
[*sequence-number*] **permit igmp** *source source-wildcard destination destination-wildcard* [*igmp-type*] [**precedence** *precedence*] [**dscp** *value*] [**fragments**]

User Datagram Protocol (UDP)
[*sequence-number*] **permit udp** *source source-wildcard* [*operator* {*portprotocol-port*}] *destination destination-wildcard* [*operator* {*portprotocol-port*}] [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**]

Syntax Description

sequence-number	(Optional) Number of the **permit** statement in the access list. This number determines the order of the statements in the access list. Range is 1 to 2147483644. (By default, the first statement is number 10, and the subsequent statements are incremented by 10.)

source	Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *source* combination as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
source-wildcard	Wildcard bits to be applied to the source. There are three alternative ways to specify the source wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *source* combination as an abbreviation for a *source* and *source-wildcard* of *source* 0.0.0.0.
protocol	Name or number of an IP protocol. It can be one of the keywords **ahp** , **esp** , **gre** , **icmp** , **igmp** , **igrp** , **ip** , **ipinip** , **nos** , **ospf** , **pim** , **pcp** , **tcp** , or **udp** , or an integer from 0 to 255 representing an IP protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the **ip** keyword. ICMP, and TCP allow further qualifiers, which are described later in this table. **Note** Filtering on AHP protocol is not supported.

destination	Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination: • Use a 32-bit quantity in four-part dotted-decimal format. • Use the **any** keyword as an abbreviation for the *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *destination* combination as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
destination-wildcard	Wildcard bits to be applied to the destination. There are three alternative ways to specify the destination wildcard: • Use a 32-bit quantity in four-part dotted-decimal format. Place ones in the bit positions you want to ignore. • Use the **any** keyword as an abbreviation for a *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255. • Use the **host** *destination* combination as an abbreviation for a *destination* and *destination-wildcard* of *destination* 0.0.0.0.
nexthop1, nexthop2, nexthop3	Specifies the next hop for this entry. **Note** You must specify the VRF for all nexthops unless the nexthop is in the default VRF.

precedence *precedence*	(Optional) Packets can be filtered by precedence level (as specified by a number from 0 to 7) or by the following names: • **Routine** —Match packets with routine precedence (0) • **priority** —Match packets with priority precedence (1) • **immediate** —Match packets with immediate precedence (2) • **flash** —Match packets with flash precedence (3) • **flash-override** —Match packets with flash override precedence (4) • **critical** —Match packets with critical precedence (5) • **internet** —Match packets with internetwork control precedence (6) • **network** —Match packets with network control precedence (7)

dscp *dscp*	(Optional) Differentiated services code point (DSCP) provides quality of service control. The values for *dscp* are as follows: • 0–63—Differentiated services codepoint value • af11—Match packets with AF11 dscp (001010) • af12—Match packets with AF12 dscp (001100) • af13—Match packets with AF13 dscp (001110) • af21—Match packets with AF21 dscp (010010) • af22—Match packets with AF22 dscp (010100) • af23—Match packets with AF23 dscp (010110) • af31—Match packets with AF31 dscp (011010) • af32—Match packets with AF32 dscp (011100) • af33—Match packets with AF33 dscp (011110) • af41—Match packets with AF41 dscp (100010) • af42—Match packets with AF42 dscp (100100) • af43–Match packets with AF43 dscp (100110) • cs1—Match packets with CS1 (precedence 1) dscp (001000) • cs2—Match packets with CS2 (precedence 2) dscp (010000) • cs3—Match packets with CS3 (precedence 3) dscp (011000) • cs4—Match packets with CS4 (precedence 4) dscp (100000) • cs5—Match packets with CS5 (precedence 5) dscp (101000) • cs6—Match packets with CS6 (precedence 6) dscp (110000) • cs7—Match packets with CS7 (precedence 7) dscp (111000) • default—Default DSCP (000000) • ef—Match packets with EF dscp (101110)

dscp range *dscp dscp*	(Optional) Differentiated services code point (DSCP) provides quality of service control. The values for *dscp* are as follows: • 0–63—Differentiated services codepoint value • af11—Match packets with AF11 dscp (001010) • af12—Match packets with AF12 dscp (001100) • af13—Match packets with AF13 dscp (001110) • af21—Match packets with AF21 dscp (010010) • af22—Match packets with AF22 dscp (010100) • af23—Match packets with AF23 dscp (010110) • af31—Match packets with AF31 dscp (011010) • af32—Match packets with AF32 dscp (011100) • af33—Match packets with AF33 dscp (011110) • af41—Match packets with AF41 dscp (100010) • af42—Match packets with AF42 dscp (100100) • af43–Match packets with AF43 dscp (100110) • cs1—Match packets with CS1 (precedence 1) dscp (001000) • cs2—Match packets with CS2 (precedence 2) dscp (010000) • cs3—Match packets with CS3 (precedence 3) dscp (011000) • cs4—Match packets with CS4 (precedence 4) dscp (100000) • cs5—Match packets with CS5 (precedence 5) dscp (101000) • cs6—Match packets with CS6 (precedence 6) dscp (110000) • cs7—Match packets with CS7 (precedence 7) dscp (111000) • default—Default DSCP (000000) • ef—Match packets with EF dscp (101110)

fragments	(Optional) Causes the software to examine noninitial fragments of IPv4 packets when applying this access list entry. When this keyword is specified, fragments are subject to the access list entry.
log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) **Note** ACL logging is supported only in ingress direction for both IPv4 and IPv6. The message includes the access list number, whether the packet was permitted or denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches a flow, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.
log-input	(Optional) Provides the same function as the **log** keyword, except that the log-message also includes the input interface.
ttl	(Optional) Turns on matching against time-to-life (TTL) value.
ttl value [*value1 ... value2*]	(Optional) TTL value used for filtering. Range is 1 to 255. If only *value* is specified, the match is against this value. If both *value1* and *value2* are specified, the packet TTL is matched against the range of TTLs between *value1* and *value2* .
icmp-type	(Optional) ICMP message type for filtering ICMP packets. Range is from 0 to 255.

icmp-code	(Optional) ICMP message code for filtering ICMP packets. Range is from 0 to 255.
igmp-type	(Optional) IGMP message type (0 to 15) or message name for filtering IGMP packets, as follows: • dvmrp • host-query • host-report • mtrace • mtrace-response • pim • precedence • trace • v2-leave • v2-report • v3-report
operator	(Optional) Operator is used to compare source or destination ports. Possible operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source* and *source-wildcard* values, it must match the source port. If the operator is positioned after the *destination* and *destination-wildcard* values, it must match the destination port. If the operator is positioned after the **ttl** keyword, it matches the TTL value. The **range** operator requires two port numbers. All other operators require one port number.
port	Decimal number a TCP or UDP port. Range is 0 to 65535. TCP ports can be used only when filtering TCP. UDP ports can be used only when filtering UDP.

protocol-port	Name of a TCP or UDP port. TCP and UDP port names are listed in the "Usage Guidelines" section. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
established	(Optional) For the TCP protocol only: Indicates an established connection.
match-any	(Optional) For the TCP protocol only: Filters on any combination of TCP flags.
match-all	(Optional) For the TCP protocol only: Filters on all TCP flags.
+ | -	(Required) For the TCP protocol **match-any** , **match-all** : Prefix *flag-name* with + or - . Use the + *flag-name* argument to match packets with the TCP flag set. Use the - *flag-name* argument to match packets when the TCP flag is not set.
flag-name	(Optional) For the TCP protocol **match-any** , **match-all** . Flag names are: **ack** , **fin** , **psh** , **rst** , **syn** , **urg** .

Command Default

ICMP message generation is enabled by default.

Command Modes

IPv4 access list configuration

Command History

Release	Modification
Release 7.0.12	This command was introduced.
Release 7.8.1	**log-input** keyword was introduced.
Release 7.5.4	**bitmask** keyword was introduced.

Usage Guidelines

Use the **permit** command following the **ipv4 access-list** command to specify conditions under which a packet can pass the access list.

By default, the first statement in an access list is number 10, and the subsequent statements are incremented by 10.

You can add **permit**, **deny**, or **remark** statements to an existing access list without retyping the entire list. To add a new **s**tatement anywhere other than at the end of the list, create a new statement with an appropriate entry number that falls between two existing entry numbers to indicate where it belongs.

Note If any ACE in an ACL contains ABF clause, this ACL cannot be applied at any non-zero compression level.

The following is a list of precedence names:

- critical
- flash
- flash-override
- immediate
- internet
- network
- priority
- routine

The following is a list of ICMP message type names:

- administratively-prohibited
- alternate-address
- conversion-error
- dod-host-prohibited
- dod-net-prohibited
- echo
- echo-reply
- general-parameter-problem
- host-isolated
- host-precedence-unreachable
- host-redirect
- host-tos-redirect
- host-tos-unreachable
- host-unknown
- host-unreachable
- information-reply
- information-request
- mask-reply
- mask-request
- mobile-redirect
- net-redirect
- net-tos-redirect
- net-tos-unreachable
- net-unreachable
- network-unknown
- no-room-for-option
- option-missing
- packet-too-big

- parameter-problem
- port-unreachable
- precedence-unreachable
- protocol-unreachable
- reassembly-timeout
- redirect
- router-advertisement
- router-solicitation
- source-quench
- source-route-failed
- time-exceeded
- timestamp-reply
- timestamp-request
- traceroute
- ttl-exceeded
- unreachable

The following is a list of TCP port names that can be used instead of port numbers. Refer to the current *Assigned Numbers* RFC to find a reference to these protocols. You can find port numbers corresponding to these protocols by typing a ? in the place of a port number.

- bgp
- chargen
- cmd
- daytime
- discard
- domain
- echo
- exec
- finger
- ftp
- ftp-data
- gopher
- hostname
- ident
- irc
- klogin
- kshell
- login
- lpd
- nntp
- pim-auto-rp
- pop2
- pop3
- smtp
- sunrpc
- tacacs

- talk
- telnet
- time
- uucp
- whois
- www

The following UDP port names can be used instead of port numbers. Refer to the current *Assigned Numbers* RFC to find a reference to these protocols. You can find port numbers corresponding to these protocols by typing a ? in the place of a port number.

- biff
- bootpc
- bootps
- discard
- dnsix
- domain
- echo
- isakmp
- mobile-ip
- nameserver
- netbios-dgm
- netbios-ns
- netbios-ss
- ntp
- pim-auto-rp
- rip
- snmp
- snmptrap
- sunrpc
- syslog
- tacacs
- talk
- tftp
- time
- who
- xdmcp

Use the following flags in conjunction with the **match-any** and **match-all** keywords and the + and - signs to select the flags to display:

- ack
- fin
- psh
- rst
- syn

For example, **match-all** +*ack* +*syn* displays TCP packets with both the ack *and* syn flags set, or **match-any** +*ack* – - *syn* displays the TCP packets with the ack set *or* the syn not set.

Task ID

Task ID	Operations
ipv4	read, write
acl	read, write

Examples

The following example shows how to set a permit condition for an access list named Internetfilter:

```
Router(config)# ipv4 access-list Internetfilter
Router(config-ipv4-acl)# 10 permit 192.168.34.0 0.0.0.255
Router(config-ipv4-acl)# 20 permit 172.16.0.0 0.0.255.255
Router(config-ipv4-acl)# 25 permit tcp host 172.16.0.0 eq bgp host 192.168.202.203 range
1300 1400
Router(config-ipv4-acl)# deny 10.0.0.0 0.255.255.255
```

This example shows how you can configure DSCP bitmask on ingress ERSPAN.

```
Router# config
Router(config)# ipv4 access-list acl1
Router(config-ipv4-acl)# 10 permit ipv4 host 192.0.2.1 any dscp af22 bitmask 0x3f
Router(config-ipv4-acl)# commit
Router(config-ipv4-acl)# exit
Router(config)# interface HundredGigE0/0/0/6
Router(config-if)# ipv4 address 192.0.2.51 255.255.255.0
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv4 acl1
Router(config-if)# commit
```

permit (IPv6)

To set permit conditions for an IPv6 access list, use the **permit** command in IPv6 access list configuration mode. To remove the permit conditions, use the **no** form of this command.

[sequence-number] **permit** *source* { *source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length* } [*operator* { *port* | *protocol-port* }] [**dscp** *value* [**bitmask** *value*]] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log | log-input**]
[sequence-number] **permit** *protocol* { *source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length* } { *source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address* } [*operator* { *port* | *protocol-port* }] [**dscp** *value* [**bitmask** *value*]] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log | log-input**]
[`ttl` *ttl* *value* [*value1* **. . .** *value2*]]
no *sequence-number*

Internet Control Message Protocol (ICMP)
[sequence-number] **permit icmp** {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length*} {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address* } {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/prefix-length*} [*icmp-type*] [*icmp-code*] [**dscp** *value*] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**log**]

Transmission Control Protocol (TCP)
[sequence-number] **permit tcp** {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/prefix-length*} [*operator* {*port* | *protocol* | *port*}] [**dscp** *value*] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**established**] {**match-any | match-all | + | -**} [*flag-name*] [**log**]

User Datagram Protocol (UDP)
[sequence-number] **permit tcp** {*source-ipv6-prefix/ prefix-length* | *any* | *host source-ipv6-address ipv6-wildcard-mask/prefix-length*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/ prefix-length* | *any* | *host destination-ipv6-address ipv6-wildcard-mask/prefix-length*} [*operator* {*port* | *protocol* | *port*}] [**dscp** *value*] [**routing**] [**hop-by-hop**] [**authen**] [**destopts**] [**fragments**] [**established**] [*flag-name*] [**log**]

Syntax Description

sequence-number	(Optional) Number of the **permit** statement in the access list. This number determines the order of the statements in the access list. Range is from 1 to 2147483644. (By default, the first statement is number 10, and the subsequent statements are incremented by 10.)

protocol	Name or number of an Internet protocol. It can be one of the keywords **ahp**, **esp**, **gre** , **icmp**, **igmp**, **igrp**, **isinip**, **ipv6**, **nos**, **ospf**, **pcp**, **sctp**, **tcp**, or **udp**, or an integer that ranges from 0 to 255, representing an IPv6 protocol number.
source-ipv6-prefix / *prefix-length*	Source IPv6 network or class of networks about which permit conditions are to be set. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
any	An abbreviation for the IPv6 prefix ::/0.
host *source-ipv6-address*	Source IPv6 host address about which to set permit conditions. This *source-ipv6-address* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
ipv6-wildcard-mask	IPv6 wildcard mask. The IPv6 wildcard mask can take any IPv6 address value which is used instead of prefix length.

operator {*port* | *protocol-port*}	(Optional) Operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range). If the operator is positioned after the *source-ipv6-prefix / prefix-length* argument, it must match the source port. If the operator is positioned after the *destination-ipv6-prefix / prefix-length* argument, it must match the destination port. The **range** operator requires two port numbers. All other operators require one port number. The *port* argument is the decimal number of a TCP or UDP port. A port number is a number whose range is from 0 to 65535. The *protocol-port* argument is the name of a TCP or UDP port. TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
destination-ipv6-prefix / prefix-length	Destination IPv6 network or class of networks about which permit conditions are to be set. This argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.
host *destination-ipv6-address*	Specifies the destination IPv6 host address about which permit conditions are to be set. This *destination-ipv6-address* argument must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.

dscp *value*	(Optional) Matches a differentiated services code point (DSCP) value against the traffic class value in the Traffic Class field of each IPv6 packet header. Range is from 0 to 63.
routing	(Optional) Matches source-routed packets against the routing extension header within each IPv6 packet header.
hop-by-hop	(Optional) Supports Jumbo-grams. With the Router Alert option, it is an integral part in the operation of Multicast Listener Discovery (MLD). Router Alert [3] is an integral part in the operations of IPv6 Multicast through MLD and RSVP for IPv6.
authen	(Optional) Matches if the IPv6 authentication header is present.
destopts	(Optional) Matches if the IPv6 destination options header is present.
fragments	(Optional) Matches noninitial fragmented packets where the fragment extension header contains a nonzero fragment offset. The **fragments** keyword is an option available only if the *operator* [*port-number*] arguments are not specified.

log	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the **logging console** command.) **Note** ACL logging is supported only in ingress direction for both IPv4 and IPv6. The message includes the access list name and sequence number, and whether the packet is permitted; the protocol, and whether it is TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first matching packet, and then at 5-minute intervals, including the number of packets permitted in the prior 5-minute interval.
log-input	(Optional) Provides the same function as the **log** keyword, except that the log-message also includes the input interface.
ttl	(Optional) Turns on matching against time-to-life (TTL) value. For IPv6 packets, **ttl** is also referred to as hop limit.
ttl value [*value1* ... *value2*]	(Optional) TTL value used for filtering. Range is 1 to 255. If only *value* is specified, the match is against this value. If both *value1* and *value2* are specified, the packet TTL is matched against the range of TTLs between *value1* and *value2* .
operator	(Optional) Operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).

icmp-type	(Optional) ICMP message type for filtering ICMP packets. Range is from 0 to 255.
icmp-code	(Optional) ICMP message code for filtering ICMP packets. Range is from 0 to 255.
established	(Optional) For the TCP protocol only: Indicates an established connection.
match-any	(Optional) For the TCP protocol only: Filters on any combination of TCP flags.
match-all	(Optional) For the TCP protocol only: Filters on all TCP flags.
`+ | -`	(Required) For the TCP protocol **match-any** , **match-all** : Prefix *flag-name* with `+` or `-` . Use the + *flag-name* argument to match packets with the TCP flag set. Use the - *flag-name* argument to match packets when the TCP flag is not set.
flag-name	(Required) For the TCP protocol **match-any**, **match-all**. Flag names are: **ack**, **fin**, **psh**, **rst**, **syn**, **urg**.

Command Default

ICMP message generation is enabled by default.

Command Modes

IPv6 access list configuration

Command History

Release	Modification
Release 7.0.12	This command was introduced.
Release 7.2.1	Ingress IPv6 TCP flags are supported.
Release 7.3.15	Egress IPv6 TCP flags are supported.
Release 7.8.1	**log-input** keyword was introduced.
Release 7.8.1	**ttl** keyword was introduced.
Release 7.5.4	**bitmask** keyword was introduced.

Release	Modification
Release 7.10.1	IPv6 AHP and ESP headers are supported.

Usage Guidelines

The **permit** (IPv6) command is similar to the **permit** (IPv4) command, except that it is IPv6-specific.

Use the **permit** (IPv6) command following the **ipv6 access-list** command to define the conditions under which a packet passes the access list.

Specifying **ipv6** for the *protocol* argument matches against the IPv6 header of the packet.

By default, the first statement in an access list is number 10, and the subsequent statements are numbered in increments of 10.

You can add **permit**, **deny, or remark** statements to an existing access list without retyping the entire list. To add a new statement anywhere other than the end of the list, create a new statement with an appropriate entry number that falls between two existing entry numbers to indicate where it belongs.

Both the *source-ipv6-prefix/prefix-length* and *destination-ipv6-prefix/prefix-length* arguments are used for traffic filtering (the source prefix filters traffic based upon the traffic source; the destination prefix filters traffic based upon the traffic destination).

Note IPv6 prefix lists, and not access lists, should be used for filtering routing protocol prefixes.

The **fragments** keyword is an option available only if the *operator* [*port* | *protocol-port*] arguments are not specified.

Task ID

Task ID	Operations
acl	read, write

Examples

This example shows how to configure the IPv6 access list named v6-abf-acl and apply the access list to inbound traffic on HundredGigE interface 0/0/2/0.

```
Router(config)# ipv6 access-list v6-abf-acl
Router(config-ipv6-acl)# 10 permit ipv6 any any
Router(config-ipv6-acl)# 20 permit ipv4 any any
Router(config)# interface HundredGigE 0/0/2/0
Router(config-if)# ipv6 access-group v6-abf-acl ingress
```

The following example shows how to configure the IPv6 access list named toCISCO and apply the access list to the traffic entering theHundredGigE interface 0/2/0/2. Specifically, the permit entry in the list allows all packets that have a hop-by-hop optional field from entering the HundredGigE interface 0/2/0/2.

```
Router(config)# ipv6 access-list toCISCO
Router(config-ipv6-acl)# permit ipv6 any any hop-by-hop
Router(config)#  interface HundredGigE 0/2/0/2
Router(config-if)# ipv6 access-group tOCISCO ingress
```

The following example shows how you can configure DSCP bitmask on ingress ERSPAN.

```
Router# config
Router(config)# ipv6 access-list acl1
Router(config-ipv6-acl)# 10 permit ipv6 host 2001:DB8::2/32 any dscp 33 bitmask 0x3f
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit
Router(config)# interface HundredGigE 0/0/10/3
Router(config-if)# ipv6 address 2001:DB8::1/32
Router(config-if)# monitor-session TEST ethernet direction rx-only port-level acl ipv6 acl1
Router(config-if)# commit
```

The following example shows how you can configure AHP and ESP headers on an ACLs.

```
Router(config)# #ipv6 access-list ipv6_umpp_access_list
Router(config-ipv6-acl)# 12 permit ahp any any
Router(config-ipv6-acl)# ipv6 access-list ipv6_umpp_access_list
Router(config-ipv6-acl)# 14 permit esp any any
Router(config-ipv6-acl)# commit
Router(config-ipv6-acl)# exit
```

show access-lists ipv4

To display the contents of current IPv4 access lists, use the **show access-lists ipv4** command in XR EXEC mode.

show access-lists ipv4 [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id* | [**usage pfilter** { **location** *node-id* }]}]

Syntax Description

access-list-name	(Optional) Name of a particular IPv4 access list. The name cannot contain spaces or quotation marks, but can include numbers.
hardware	(Optional) Identifies the access list as an access list for an interface.
ingress	(Optional) Specifies an inbound interface.
interface	(Optional) Displays interface statistics.
type	(Optional) Interface type. For more information, use the question mark (?) online help function.
interface-path-id	Physical interface or virtual interface. **Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router. For more information about the syntax for the router, use the question mark (?) online help function.
sequence *number*	(Optional) Sequence number of a particular IPv4 access list. Range is 1 to 2147483644.
location *node-id*	(Optional) Location of a particular IPv4 access list. The *node-id* argument is entered in the *rack/slot/module* notation.
summary	(Optional) Displays a summary of all current IPv4 access lists.

sequence-number	(Optional) Sequence number of a particular IPv4 access list. Range is 1 to 2147483644.
usage	(Optional) Displays the usage of the access list on a given line card.
pfilter	(Optional) Displays the packet filtering usage for the specified line card.

Command Default

The default displays all IPv4 access lists.

Command Modes

XR EXEC mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

Use the **show access-lists ipv4** command to display the contents of all IPv4 access lists. To display the contents of a specific IPv4 access list, use the *name* argument. Use the *sequence-number* argument to specify the sequence number of the access list.

Use the **hardware , ingress** and **location** keywords to display the access list hardware contents and counters for all interfaces that use the specified access list in a given direction. To display the contents of a specific access list entry, use the **sequence** *number* keyword and argument. The access group for an interface must be configured using the **ipv4 access-group** command for access list hardware counters to be enabled.

Use the **show access-lists ipv4 summary** command to display a summary of all current IPv4 access lists. To display a summary of a specific IPv4 access list, use the *name* argument.

Use the **show access-list ipv4 usage** command to display a summary of all interfaces and access lists programmed on the specified line card.

Task ID

Task ID	Operations
acl	read

Examples

In the following example, the contents of all IPv4 access lists are displayed:

```
Router# show access-lists ipv4

ipv4 access-list test_ipv4
 10 permit ipv4 any any
 20 deny tcp any eq 2000 any eq 2000
 30 permit tcp any eq 3000 any eq 3000
```

This table describes the significant fields shown in the display.

Table 1: show access-lists ipv4 hardware Field Descriptions

Field	Description
hw matches	Number of hardware matches.
ACL name	Name of the ACL programmed in hardware.
Sequence Number	Each ACE sequence number is programmed into hardware with all the fields that are corresponding to the values set in ACE.
Grant	Depending on the ACE rule, the grant is set to deny, permit, or both.
Logging	Logging is set to on if ACE uses a log option to enable logs.
Per ace icmp	If Per ace icmp is set to on in the hardware, ICMP is unreachable, is rate-limited, and is generated. The default is set to on.
Hits	Hardware counter for that ACE.

In the following example, a summary of all IPv4 access lists are displayed:

```
Router# show access-lists ipv4 summary

ACL Summary:
  Total ACLs configured: 3
  Total ACEs configured: 11
```

This table describes the significant fields shown in the display.

Table 2: show access-lists ipv4 summary Field Descriptions

Field	Description
Total ACLs configured	Number of configured IPv4 ACLs.
Total ACEs configured	Number of configured IPV4 ACEs.

This example displays the packet filtering usage for the specified line card:

```
Router# show access-lists ipv4 usage pfilter location 0/RP0/CPU0

 Interface : HundredGigE0/0/0/10/0
Input ACL : Common-ACL : N/A ACL : test_ipv4
Output ACL : N/A
```

Note To display the packet filtering usage for bundle interfaces, use the **show access-lists ipv4 usage pfilter location all** command.

show access-lists ipv6

To display the contents of current IPv6 access lists, use the **show access-lists ipv6** command in XR Config mode.

show access-lists ipv6 [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id* | [**usage pfilter** { **location** *node-id* }]}]

Syntax Description

access-list-name	(Optional) Name of a particular IPv6 access list. The name cannot contain a spaces or quotation marks, but can include numbers.
hardware	(Optional) Identifies the access list as an access list for an interface.
ingress	(Optional) Specifies an inbound interface.
interface	(Optional) Displays interface statistics.
type	(Optional) Interface type. For more information, use the question mark (?) online help function.
interface-path-id	(Optional) Either a physical interface instance or a virtual interface instance as follows: • Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation. • *rack*: Chassis number of the rack. • *slot*: Physical slot number of the modular services card or line card. • *module*: Module number. A physical layer interface module (PLIM) is always 0. • *port*: Physical port number of the interface. **Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0) and the module is CPU0. Example: interface MgmtEth0/RSP0/CPU0/0. • Virtual interface instance. Number range varies depending on interface type. For more information about the syntax for the router, use the question mark (?) online help function.
sequence *number*	(Optional) Sequence number of a particular IPv6 access list. Range is 1 to 2147483644.
location *node-id*	(Optional) Location of a particular IPv6 access list. The *node-id* argument is entered in the *rack/slot/module* notation.
summary	(Optional) Displays a summary of all current IPv6 access lists.
sequence-number	(Optional) Sequence number of a particular IPv6 access list. Range is 1 to 2147483644.
usage	(Optional) Displays the usage of the access list on a given line card.

pfilter	(Optional) Displays the packet filtering usage for the specified line card.
all	(Optional) Displays the location of all the line cards.

Command Default

Displays all IPv6 access lists.

Command Modes

XR Config mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

The **show access-lists ipv6** command is similar to the **show access-lists ipv4** command, except that it is IPv6 specific.

Use the **show access-lists ipv6** command to display the contents of all IPv6 access lists. To display the contents of a specific IPv6 access list, use the *name* argument. Use the *sequence-number* argument to specify the sequence number of the access list.

Use the **hardware , ingress** and **location** keywords to display the access list hardware contents and counters for all interfaces that use the specified access list in a given direction. To display the contents of a specific access list entry, use the **sequence** *number* keyword and argument. The access group for an interface must be configured using the **ipv6 access-group** command for access list hardware counters to be enabled.

Use the **show access-lists ipv6 summary** command to display a summary of all current IPv6 access lists. To display a summary of a specific IPv6 access list, use the *name* argument.

Use the **show access-list ipv6 usage** command to display a summary of all interfaces and access lists programmed on the specified line card.

Task ID

Task ID	Operations
acl	read

Examples

In the following example, the IPv6 ACL is configured with the source IPv6 wildcard mask FF:0:FFFF:AA:20 and the destination wildcard mask 0:FFFF:2233::FFFF, the show command displays these wildcard mask:

```
Router# config
Router(config)# ipv6  access-list acl1
Router(config-ipv6-acl)# permit 1:2::3 FF:0:FFFF:AA:20:: 4:5::6 0:FFFF:2233::FFFF
Router(config-ipv6-acl)# commit
Router# show run ipv6 access-list
ipv6 access-list ACL1
 10 permit ipv6 1:2::3 ff:0:ffff:aa:20:: 4:5::6 0:ffff:2233::ffff
```

In the following example, the contents of all IPv6 access lists are displayed:

```
Router# show access-lists ipv6

ipv6 access-list test_ipv6
```

```
 10 permit ipv6 any any
 20 permit tcp any eq 3000 any eq 3000
```

In the following example, the contents of an access list named Internetfilter is displayed:

```
Router# show access-lists ipv6 Internetfilter

ipv6 access-list Internetfilter
  3 remark Block BGP traffic from a given host
  4 deny tcp host 6666:1:2:3::10 eq bgp host 7777:1:2:3::20 range 1300 1404 deny tcp host
171.69.2.88 255.255.0.0 any eq telnet
  20 permit ipv6 3333:1:2:3::/64 any
  25 permit ipv6 4444:1:2:3::/64 any
  30 permit ipv6 5555:1:2:3::/64 any
```

This table describes the significant fields shown in the display.

Table 3: show access-lists ipv6 hardware Command Field Descriptions

Field	Description
hw matches	Number of hardware matches.

In the following example, a summary of all IPv6 access lists is displayed:

```
Router# show access-lists ipv6 summary

ACL Summary:
  Total ACLs configured: 3
  Total ACEs configured: 11
```

This table describes the significant fields shown in the display.

Table 4: show access-lists ipv6 summary Command Field Descriptions

Field	Description
Total ACLs configured	Number of configured IPv6 ACLs.
Total ACEs configured	Number of configured IPV6 ACEs.

In the following example, the OOR details of the IPv6 access lists are displayed:

```
Router# show access-lists ipv6 maximum detail

Default max configurable acls :1000
Default max configurable aces :50000
Current configured acls       :1
Current configured aces       :2
Current max configurable acls :1000
Current max configurable aces :50000
Max configurable acls         :2000
Max configurable aces         :100000
```

This example displays the packet filtering usage for the specified line card:

```
Router# show access-lists ipv6 usage pfilter location 0/0/CPU0

Interface : HundredGigE0/0/0/10/0
    Input  ACL : Common-ACL : N/A  ACL : test_ipv6
    Output ACL : N/A
```

This example displays the ABF ACL match counter statistics when the **show access-lists ipv6** command is used with the **hardware** option.

```
Router# show access-lists ipv6 abf_1_v6 hardware ingress location 0/RP0/CPU0

Fri Sep 27 17:18:19.288 PDT
ipv6 access-list abf_1_v6
 5 permit ipv6 any 2001:150:25:207::/64 (1883512876 matches) (next-hop:
addr=2620:149:bb:3210::90, vrf name=default)
 10 permit ipv6 any any (153796235 matches)
```

show tech-support access-lists

To automatically collect information about Ethernet Services, IPV4, IPV6, and Platform dependent ACL related information, use the **show tech-support access-lists** command in configuration mode.

show tech-support access-lists { **ethernet-services** | **ipv4** | **ipv6** | **platform** }

Syntax Description

ethernet-services	Collects information regarding the ethernet-services access lists in the router.
ipv4	Collects information regarding the ipv4 access lists in the router.
ipv6	Collects information regarding the ipv6 access lists in the router.
platform	Collects information regarding the platform specific access lists in the router.

Command Default

None

Command Modes

Configuration mode

Command History

Release	Modification
Release 7.0.12	This command was introduced.

Usage Guidelines

- To use commands, you must be in a user group associated with a task group that includes appropriate task IDs. If the user group assignment is preventing you from using any command, contact your AAA administrator for assistance.
- This command generates tech-support information that is useful for Cisco Technical Support representatives when troubleshooting a router. By default, the output of this command is saved on the router's hard disk in a file with *.tgz* extension. You can share this file with Cisco Technical Support. To share, use the **copy** command to copy the *.tgz* file to a server or local machine. For example, **copy harddisk:/showtech/** *name.tgz* **tftp://** *server_path* .
- This command is not required during normal use of the router.

Task ID

Task ID	Operations
acl	read

Examples

The following example shows the output of the **show tech-support access-lists** command:

```
Router# show tech-support access-lists ipv4
Thu Oct 20 10:38:18.041 PDT
++ Show tech start time: 2022-Oct-20.103818.PDT ++
Thu Oct 20 10:38:18 PDT 2022 Waiting for gathering to complete
.....
Thu Oct 20 10:38:33 PDT 2022 Compressing show tech output
Show tech output available at 0/RP0/CPU0 :
/harddisk:/showtech/showtech-M8102TOR1-ipv4-acl-2022-Oct-20.103818.PDT.tgz
++ Show tech end time: 2022-Oct-20.103833.PDT ++
```

tcam format access-list (ipv4 and ipv6)

To configure the object group ACLs for IPv4 and IPv6 using the user-defined TCAM keys (UDK), use the **hw-module profile tcam format access-list ipv4** command and **hw-module profile tcam format access-list ipv6** in XR Config mode.

Syntax for IPv4:

hw-module profile tcam format access-list ipv4

Syntax Description

dst-addr	Specifies destination address. This is a 32-bit qualifier for IPv4 ACLs.
dst-object-group	Specifies the destination object group.
dst-port	Destination port for TCP/UDP. This is a 16-bit qualifier.
frag-bit	Fragmentation bit for IPv4 ACLs. This is a 1-bit qualifier.
fragment-offset	Specifies the fragment offset for IPv4 ACLs.
packet-len	Specifies packet length for IPv4 ACLs. This is a 10-bit qualifier.
precedence	Specifies DSCP precedence in IPv4 header. This is a 10-bit qualifier.
proto	Specifies protocol type in IPv4 header. This is an 8-bit qualifier.
src-addr	Specifies source address. This is a 32-bit qualifier for IPv4 ACLs.
src-object-group	Specifies the source object group.
src-port	Specifies source port for TCP/UDP. This is a 16-bit qualifier.
tcp-flags	Specifies TCP Flags. This is a 6-bit qualifier for IPv4 ACLs.

Syntax for IPv6:

hw-module profile tcam format access-list ipv6

Syntax Description

dst-addr	Specifies destination address. This is a 128-bit qualifier for IPv6 ACLs.
dst-object-group	Specifies the destination object group.
dst-port	Destination port for TCP/UDP. This is a 16-bit qualifier.
frag-bit	Fragmentation bit for IPv6 ACLs. This is a 1-bit qualifier.
next-hdr	(Mandatory) Specifies the next header field in IPv6 header. This is an 8-bit qualifier.
packet-len	Specifies packet length for IPv6 ACLs. This is a 10-bit qualifier.
src-addr	Specifies source address. This is a 128-bit qualifier for IPv6 ACLs.
src-object-group	Specifies the source object group.

src-port	(Mandatory) Specifies source port for TCP/UDP. This is a 16-bit qualifier.
tcp-flags	Specifies TCP Flags. This is an 8-bit qualifier for IPv6 ACLs.
traffic-class	Specifies traffic class in IPv6 header. This is an 8-bit qualifier for IPv6 ACLs.

Command Default

None

Command Modes

XR Config mode

Command History

Release	Modification
Release 24.2.1	These commands were introduced.

Usage Guidelines

- Remove all ACL attachments to interfaces before the IPv4/IPv6 UDK configuration.
- Make sure that you reload the line card for this configuration to take effect.

Task ID

Task ID	Operations
acl	read, write
ipv4	read, write
ipv6	read, write

Examples

Example 1: In UDK, if only the **dst-object-group** is specified and the **src-object-group** is not specified, you compress only the destination address (compress level 4) as shown in this example.

```
Router(config)# hw-module profile tcam format access-list ipv4 src-addr src-port dst-port
proto tcp-flags frag-bit dst-object-group
Router(config)# hw-module profile tcam format access-list ipv6 src-addr src-port dst-port
next-hdr frag-bit tcp-flags dst-object-group
```

```
interface FH0/0/0/1
 RP/0/RP0/CPU0:ios(config-if)#ipv6 access-group v6-test ingress compress level 4
```

Example 2: In UDK, if only the **src-object-group** is specified and the **dst-object-group** is not specified, you compress only the source address (compress level 1) as shown in this example.

```
Router(config)# hw-module profile tcam format access-list ipv4 src-object-group src-port
dst-port proto tcp-flags frag-bit dst-addr
Router(config)# hw-module profile tcam format access-list ipv6 src-object-group src-port
dst-port next-hdr frag-bit tcp-flags dst-addr
```

```
interface FH0/0/0/1
 RP/0/RP0/CPU0:ios(config-if)#ipv4 access-group v4-test ingress compress level 1
```

Note By default, compression level 2 is supported for both the **src-object-group** and **dst-object-group** without the UDK configuration.